**Business Requirement:**

Develop an Uploader by using which you can upload Master data on any Z or Y table (Custom Table).

---------------------------------------------------------------------------------

**Program Name : ZR\_MASTER\_DATA\_UPLOAD**

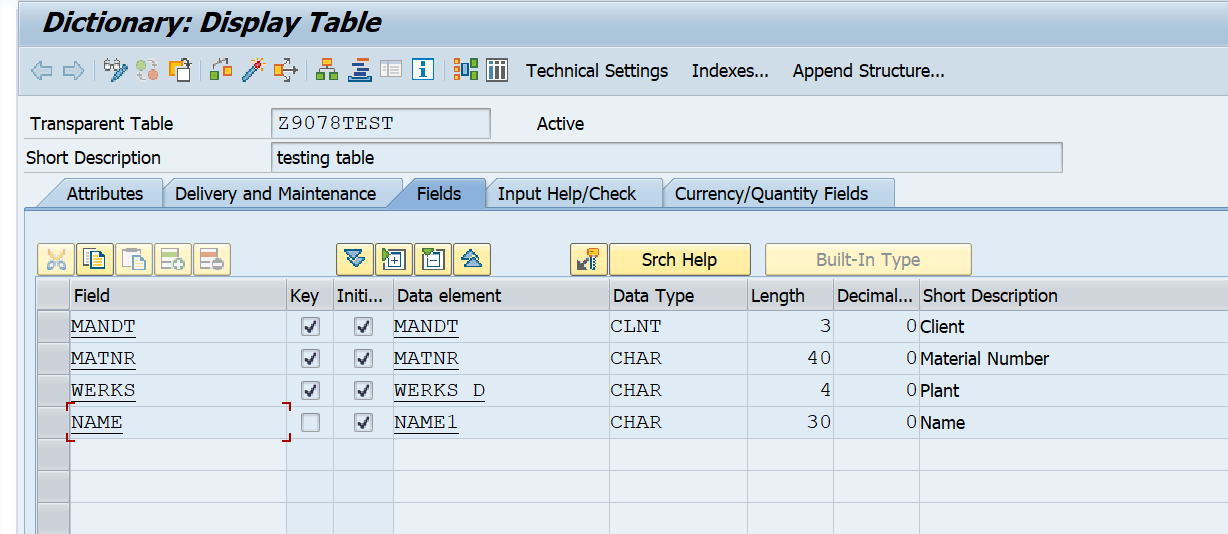
**Description: Uploading data into custom( Z ) table from Excel**

* ---------------------------------------------------------------------------------------------------------
* *\*&---------------------------------------------------------------------\**  
  *\*& Report ZR\_MASTER\_DATA\_UPLOAD*  
  *\*&---------------------------------------------------------------------\**  
  *\*&*  
  *\*&---------------------------------------------------------------------\**  
  REPORT zr\_master\_data\_upload.  
  *\*TABLES : zmm\_vend\_grp.*  
  *\*DATA:it\_final TYPE STANDARD TABLE OF zmm\_vend\_grp,*  
  *\*     wa\_final TYPE LINE OF it\_final.*  
  DATA:lt\_excel       TYPE TABLE OF alsmex\_tabline,  
       lt\_dref        TYPE REF TO data,  
       ls\_dref        TYPE REF TO data,  
       lv\_col         TYPE i,  
       lo\_alv         TYPE REF TO cl\_salv\_table,  
       lt\_table\_filds TYPE TABLE OF dfies.  
  *\*\*\* --- Field Symbols*  
  FIELD-SYMBOLS : <fs\_table> TYPE any .  
  FIELD-SYMBOLS : <ft\_table> TYPE STANDARD TABLE.  
  FIELD-SYMBOLS : <dyn\_field> .  
    
  *\*\*\* --- Selection screen designing*  
  SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME TITLE TEXT-001.  
  PARAMETERS:p\_file  TYPE rlgrap-filename OBLIGATORY,  
             p\_table TYPE dd02l-tabname OBLIGATORY,  
             p\_test  AS CHECKBOX DEFAULT abap\_true.  
  SELECTION-SCREEN END OF BLOCK b1.  
    
  AT SELECTION-SCREEN ON p\_table.  
    IF p\_table+0(1) NE 'Z'.  
      MESSAGE 'You cannot update standard table.' TYPE 'E'.  
    ENDIF.  
    
  *\*\*\* --- value request for p\_file*  
  AT SELECTION-SCREEN ON VALUE-REQUEST FOR p\_file.  
    CALL FUNCTION 'F4\_FILENAME'  
      IMPORTING  
        file\_name = p\_file.  
  *\*\*\* --- START-OF-SELECTION*  
  START-OF-SELECTION.  
  *\*create OBJECT lr\_descr.*  
  *\*\*\* --- Assigning fields symbols for Tables*  
    CREATE DATA lt\_dref TYPE TABLE OF (p\_table).  
    CREATE DATA ls\_dref TYPE (p\_table).  
  *\*\*\* --- Assign field symbol with table type of DDIC*  
    ASSIGN lt\_dref->\* TO <ft\_table>.  
  *\*\*\* --- Assign field symbol with Structure type of DDIC*  
    ASSIGN ls\_dref->\* TO <fs\_table>.  
  *\*\*\* --- Call the Function module ALSM\_EXCEL\_TO\_INTERNAL\_TABLE*  
    CALL FUNCTION 'ALSM\_EXCEL\_TO\_INTERNAL\_TABLE'  
      EXPORTING  
        filename                = p\_file  
        i\_begin\_col             = 1  
        i\_begin\_row             = 1  
        i\_end\_col               = 99  
        i\_end\_row               = 999999  
      TABLES  
        intern                  = lt\_excel  
      EXCEPTIONS  
        inconsistent\_parameters = 1  
        upload\_ole              = 2  
        OTHERS                  = 3.  
    IF sy-subrc EQ 0.  
  *\*\*\* --- Sort*  
      SORT lt\_excel BY row.  
    
      LOOP AT lt\_excel INTO DATA(ls\_excel).  
    
  *\*\*\* --- Adding count to skip the mapping for MANDT field*  
  *\*      lv\_col = ls\_excel-col + 1.*  
        lv\_col = ls\_excel-col .  
        ASSIGN COMPONENT lv\_col OF STRUCTURE <fs\_table> TO <dyn\_field>.  
        IF sy-subrc = 0.  
  *\*\*\*-- Compare Excel sheet column data and Dynamic table fields*  
          IF ls\_excel-row = 1.  
            CALL FUNCTION 'DDIF\_FIELDINFO\_GET'  
              EXPORTING  
                tabname        = p\_table  
              TABLES  
                dfies\_tab      = lt\_table\_filds  
              EXCEPTIONS  
                not\_found      = 1  
                internal\_error = 2  
                OTHERS         = 3.  
            READ TABLE lt\_table\_filds INTO DATA(ls\_table\_fields) INDEX lv\_col.  
            IF sy-subrc = 0.  
              IF ls\_table\_fields-fieldname NE ls\_excel-value.  
                WRITE: 'Excel sheet data and Dynamic table fields are not matching'.  
                EXIT.  
              ENDIF.  
            ENDIF.  
          ELSE.  
            IF ls\_excel-value+2(1) = '.' AND  ls\_excel-value+5(1) = '.' OR  
               ls\_excel-value+2(1) = '/' AND  ls\_excel-value+5(1) = '/'.  
              CONCATENATE ls\_excel-value+6(4) ls\_excel-value+3(2) ls\_excel-value+0(2) INTO <dyn\_field>.  
            ELSE.  
              <dyn\_field> = ls\_excel-value.  
            ENDIF.  
          ENDIF.  
        ENDIF.  
        IF ls\_excel-row GT 1.  
          AT END OF row.  
            APPEND <fs\_table> TO <ft\_table>.  
            CLEAR <fs\_table>.  
          ENDAT.  
        ENDIF.  
      ENDLOOP.  
    
      IF <ft\_table> IS NOT INITIAL.  
        IF p\_test IS INITIAL.  
  *\*\*\* --- Modify*  
          MODIFY (p\_table) FROM TABLE <ft\_table>.  
          IF sy-subrc EQ 0.  
            COMMIT WORK.  
            DATA(lv\_lines) = lines( <ft\_table> ).  
            WRITE: TEXT-002,lv\_lines.  
          ELSE.  
            ROLLBACK WORK.  
            MESSAGE TEXT-003 TYPE 'E' DISPLAY LIKE 'I'.  
          ENDIF.  
        ELSE.  
  *\*\*\* --- Factory Method*  
          cl\_salv\_table=>factory(  
            IMPORTING  
              r\_salv\_table   =  lo\_alv   *" Basis Class Simple ALV Tables*  
            CHANGING  
              t\_table        = <ft\_table>  
          ).  
  *\*\*\* --- Display*  
          lo\_alv->display( ).  
        ENDIF.  
      ENDIF.  
    ENDIF.

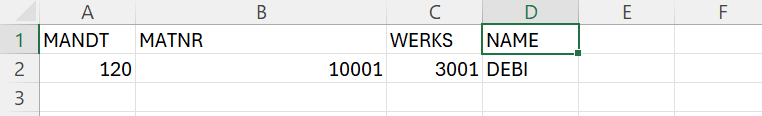
Testing Screen

Note :Here we have to take same to same technical name of Field of the database in Uploader with Sequence,

TABLE NAME : Z9078TEST

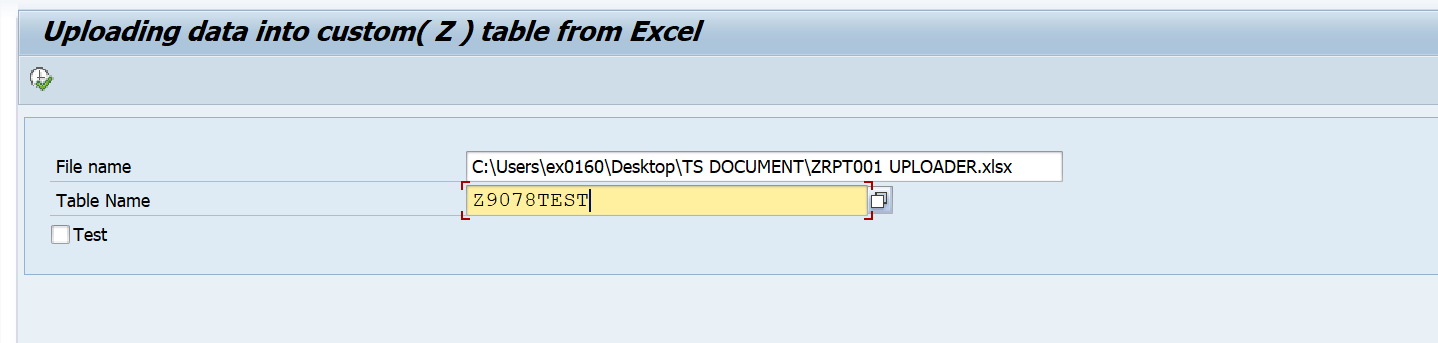


Excel

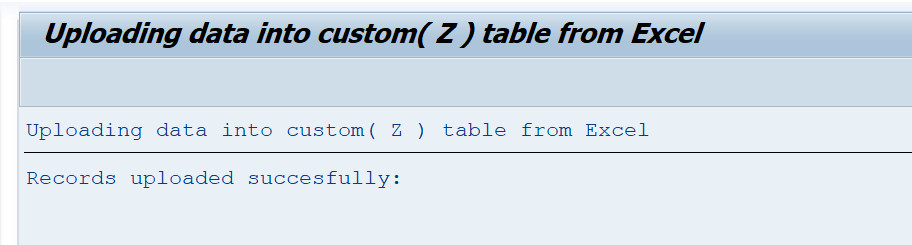


**Case 1: If Test check box is Checked.**

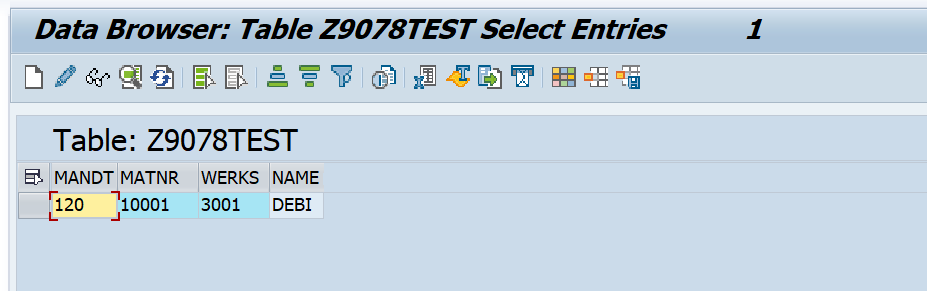
Input SCREEN



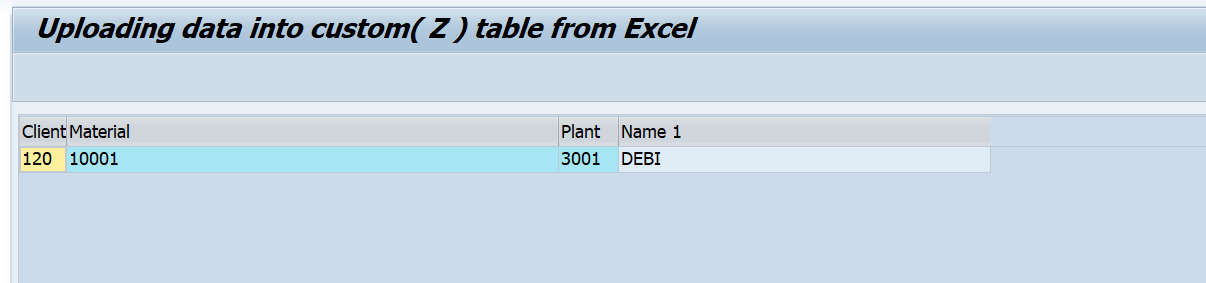
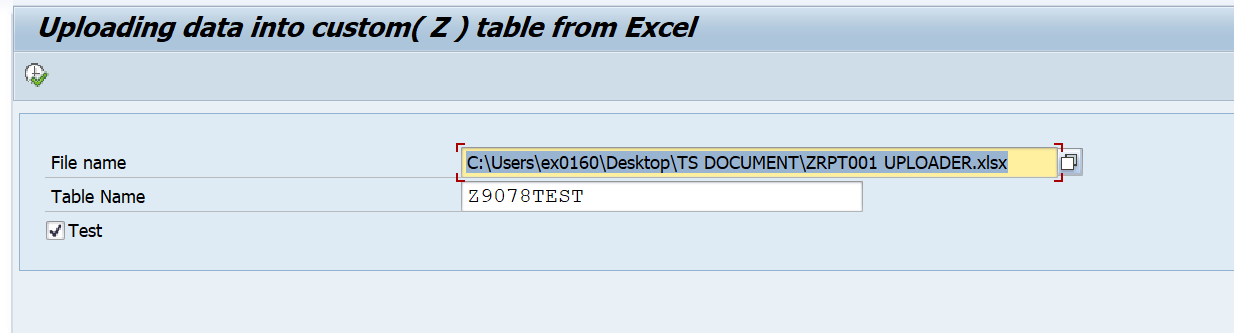
Output screen.



Record in Database.



**Case 2: If Test check box is not Checked.**



Note: **Test** Check box provided to check before upload and it’s recommended